A
Project Report
on

Smart Farming Application Using Machine Learning Algorithm

Submitted to

Sant Gadge Baba Amravati University, Amravati

Submitted in partial fulfilment of

the requirements for the Degree of

Bachelor of Engineering in

Electronics and Telecommunication Engineering

Submitted by

Shreya Gaikwad

(PRN: 193120317)

Shraddha Bhore

(PRN: 193120185)

Prem Hargunani

(PRN: 193120254)

Atharva Joshi

(PRN: 193120073)

Under the Guidance of

Dr. K. B Khanchandani

Professor, E & TC Dept.

Department of Electronics & Telecommunication Engg.

Shri Sant Gajanan Maharaj College of Engineering,

Shegaon – 444 203 (M.S.)

2022-2023

Department of Electronics & Telecommunication Engineering
Shri Sant Gajanan Maharaj College of Engineering,
Shegaon – 444203, Maharashtra, India
(Recognized by AICTE, Accredited by N.B.A, New Delhi)

Certificate

This is to certify that the project report entitled **"Precision Farming Application Using Machine Learning Algorithm"** is hereby approved as a creditable study carried out and presented by

Shreya Gaikwad (PRN: 193120317)
Shraddha Bhore (PRN: 193120185)
Prem Hargunani (PRN: 193120254)
Atharva Joshi (PRN: 193120073)

in a manner satisfactory to warrant of its acceptance as a pre-requisite in a partial fulfillment of the requirements for the degree of Bachelor of Engineering in Electronics & Telecommunication Engineering of Sant Gadge Baba Amravati University, Amravati during the Session 2022-23.

Dr. K. B. Khanchandani
Project Guide

Dr. P. R.Wankhede
Internal Examinar

31/05/23
External Examiner

Dr. M. N. Tibdewal
Professor & Head, E & TC Dept.

Dr. S. B. Somani
Principal

Abstract

In general, agriculture is the backbone of India and also plays an important role in Indian economy by providing a certain percentage of domestic product to ensure the food security. But now-a-days, food production and prediction is getting depleted due to unnatural climatic changes, which will adversely affect the economy of farmers by getting a poor yield and also help the farmers to remain less familiar in forecasting the future crops. This research work helps the beginner farmer in such a way to guide them for sowing the reasonable crops by deploying machine learning, one of the advanced technologies in crop prediction. Naive Bayes, a supervised learning algorithm puts forth in the way to achieve it. The seed data of the crops are collected here, with the appropriate parameters like temperature, humidity, and moisture content, which helps the crops to achieve a successful growth. In addition, as the software, a mobile application for Android is being developed. The users are encouraged to enter parameters like temperature and their location will be taken automatically in this application in order to start the prediction process.

Keywords — Precision Agriculture, Machine learning, Crop prediction, Naive Bayes, Supervised Learning, Effective farming.

Acknowledgement

We would like to take this opportunity to express our heartfelt thanks to our guide **Dr. K.B. Khanchandani** for his esteemed guidance and encouragement, especially through difficult times. His suggestions broaden our vision and guided us to succeed in this work. We are also very grateful for his guidance and comments while designing part of our project and learnt many things under his leadership. Also we would like to thank to Dr. M. N. Tibdewal, Head of Electronics and Telecommunication Department, all teaching and non-teaching staff of EXTC Department for their encouragement and suggestions for our project.

We extend our thanks to Dr. S. B. Somani, Principal, Shri Sant Gajanan Maharaj, College of Engineering, for his valuable support.

We sincerely thank to all our friends, who helped us directly or indirectly in completing our project work. We would like to express our appreciation for the wonderful experience while completions of this project work.

Shreya Gaikwad
Shraddha Bhore
Prem Hargunani
Atharva Joshi

Abbreviations

AI	-	Artificial Intelligence
SVM	-	Support Vector Machine
ML	-	Machine Learning
ANN	-	Artificial Neural Networks
RF	-	Random Forest
IEEE	-	Institute of Electrical and Electronics Engineers
IoT	-	Internet of Things
AWS IoT	-	Amazon Internet Of Things
IBM	-	International Business Machines
IRJET	-	International Research Journal of Engineering and Technology
IJCSSE	-	International Journal of Computer Science and Software Engineering
ICSTM	-	International Conference on Smart Technologies and Management

List of Figures

List of Tables

Contents

CHAPTER.1

INTRODUCTION

Chapter 1

INTRODUCTION

Technology is used in precision farming to maximize crop yields while minimizing inputs like fertilizer, water, and pesticides. A subfield of artificial intelligence known as machine learning (ML) enables computers to learn from data without being explicitly programmed. This report provides an overview of ML-based applications for precision farming, including their advantages, drawbacks, and potential future developments. Benefits of Precision Farming with ML Precision farming with ML provides farmers with the following benefits:

Improved Yield:

ML can analyze data from multiple sources such as satellite imagery, sensors, and weather stations to help farmers make informed decisions about planting, irrigation, and fertilization. By using ML, farmers can identify the optimal planting time and location, leading to higher crop yields.

Reduced Cost:

By optimizing resource usage, precision farming can reduce the cost of inputs such as fertilizer, water, and pesticides. ML can analyze historical data to identify patterns and provide recommendations that help farmers reduce waste and minimize input costs Environmental Sustainability: Precision farming with ML can help reduce environmental impact by minimizing the use of fertilizers, pesticides, and water. By using ML to analyze data from sensors, farmers can identify areas of their farm that require less irrigation and apply water only where needed, reducing water consumption. Challenges of Precision Farming with ML Despite the benefits of precision farming with ML, there are also several challenges:

Data Availability:

The quantity and quality of the data that is available are crucial factors in ML models' success. From various sources like weather stations, satellite imagery, and sensors, precision farming generates a lot of data. Notwithstanding, quite a bit of this information might be unstructured or deficient, making it challenging to use for ML models.

Model Accuracy:

ML models rely on accurate data to make predictions. However, environmental factors such as weather patterns and soil composition can vary significantly between farms, making it challenging to develop accurate models that work across different locations.

Technical Expertise:

Precision farming with ML requires specialized knowledge and technical expertise. Farmers may require training to understand how to collect and analyze data effectively and how to interpret the results. Future Developments Precision farming with ML is an area of active research and development, and there are several areas of future potential:

Edge Computing: Edge computing involves processing data close to the source, such as on the farm, rather than sending it to a centralized location. Edge computing can enable real-time data analysis, reducing latency and improving decision-making.

Automated Decision Making:

ML can be used to automate decision-making processes in precision farming. For example, ML models could be used to trigger irrigation systems automatically based on soil moisture levels or to apply pesticides only when necessary.

Data Integration: Integrating data from multiple sources such as sensors, satellite imagery, and weather stations can provide a more comprehensive view of the farm. ML can be used to analyze this data and provide farmers with insights that are not possible with a single data source.

1.1 Motivation

There are several situations which cannot be handled by humans. In farming, there are some situations such as parrots mostly attacks on the fruit at dawn and dusk, human's faces low eye sight problems, the risk involve in the walkabout of the farm and in the rainy season, farmer cannot go to the farm after heavy rainfall. By doing analysis of the above problems faced by farmers, we found that Quadcopter can be one of the best suitable solutions to overcome these issues. It can easily reach to a specific height, so low eye sight problem get solved. In rainy season, mud becomes serious issue to walkabout the farm. So, it can be easily overcome. Farmer can easily give command by sending message to the system. Longer area can be survey within few minutes.

1.2 Objectives

The objective of crop prediction applications is to help farmers make informed decisions about their crop production by providing accurate predictions about crop yields, crop diseases, weather conditions, market prices, and other relevant variables.

These applications use machine learning algorithms to analyze historical data and make predictions about future crop production, which can help farmers plan their planting schedules, choose the best crops to grow, and make other important decisions. Some of the specific objectives of crop prediction applications may include:

1. Maximizing Crop Yields: One of the main objectives of crop prediction applications is to help farmers maximize their crop yields. By predicting future crop yields based on historical data and other relevant variables, farmers can make more informed decisions about crop management practices such as irrigation, fertilization, and pest control.
2. Minimizing Crop Losses: Another objective of crop prediction applications is to help farmers minimize their crop losses due to factors such as crop diseases, weather conditions, and market prices. By predicting the occurrence of crop diseases and adverse weather conditions, farmers can take preventative measures to protect their crops and minimize losses.
3. Optimizing Planting Schedules: Crop prediction applications can also help farmers optimize their planting schedules by predicting the best time to plant different crops based on weather conditions, soil moisture levels, and other relevant variables. This can help farmers maximize their yields and reduce the risk of crop failure.
4. Making Informed Marketing Decisions: Crop prediction applications can also provide farmers with information about market prices and demand for different crops. This can help farmers make informed decisions about what crops to grow and when to sell them, which can increase their profits and reduce waste.

Overall, the objective of crop prediction applications is to help farmers make more informed decisions about their crop production by providing accurate predictions about crop yields, crop diseases, weather conditions, market prices, and other relevant variables.

1.3 System Overview

An overview of a crop prediction application could include the following:

1. User Interface: The application would have a user interface that allows the user to input the NPK values or other relevant variables. The user interface may also display the predicted crop yield, along with other relevant information such as weather forecasts, market prices, and crop disease predictions.

2. Data Input: The user would input the NPK values or other relevant variables into the application. The application would then use machine learning algorithms to analyze the input data and make predictions about crop yields, crop diseases, weather conditions, and other relevant variables.

3. Machine Learning Models: The application would use machine learning models such as Random Forest, Linear Regression, or Neural Networks to analyze the input data and make predictions. These models would be trained on historical data of crop yields and other relevant variables to improve their accuracy and make more accurate predictions.

4. Predictions: Based on the input data and the machine learning models, the application would make predictions about crop yields, crop diseases, weather conditions, and other relevant variables. These predictions would be displayed to the user through the user interface, along with other relevant information such as weather forecasts and market prices.

5. Feedback and Improvements: The application would continually receive feedback from users and monitor its predictions to identify areas where improvements can be made. This feedback would be used to improve the accuracy of the machine learning models and make more accurate predictions.

Overall, a crop prediction application would use machine learning algorithms to analyze input data and make predictions about crop yields, crop diseases, weather conditions, and other relevant variables. The application would display these predictions to the user through a user interface, and use feedback and monitoring to improve the accuracy of its predictions over time.

1.4 Outline of project

The outline provides a general framework for a project plan for a Precision Farming Application. Depending on the specifics of the project, certain sections may be expanded or modified, and additional sections may be added as needed. The project iscreated by following below steps:

- Study of literature survey
- Advantages and Disadvantages of the product
- Advantages and Disadvantages of the Application
- Deciding the methodology, Roadmap of design
- Finalization of the software
- Design development and performance analysis

Overall, developing a crop prediction application involves a combination of data analysis, machine learning, and software development. The key to success is to ensure that the application is accurate, user-friendly, and capable of providing farmers with the insights they need to optimize their crop production.

CHAPTER.2

Literature Review

Chapter 2

LITERATURE REVIEW

This section of report provides the summary of published papers and products available in the market that were studied during the whole process.

Crop Yield Prediction Using Machine Learning Algorithms:

This paper presents the Crop yield prediction in agriculture that helps farmers to make informed decisions about crop management, planting, and harvesting. Machine learning algorithms have shown promising results in crop yield prediction due to their ability to handle complex and large datasets. In this literature review, we will examine the current state-of-the-art techniques for crop yield prediction using machine learning algorithms. We conducted a systematic review of published articles on crop yield prediction using machine learning algorithms from several academic databases such as IEEE Xplore, ScienceDirect, and SpringerLink. We focused on articles published between 2018 and 2021.Several machine learning algorithms have been used for crop yield prediction, including linear regression, decision tree, random forest, support vector machine (SVM), neural networks, and deep learning models. The following are some of the studies that have used machine learning algorithms for crop yield prediction. In 2018, Raza et al. used SVM, random forest, and artificial neural network (ANN) models for predicting wheat yield in Pakistan. The SVM model performed the best, with an accuracy of 89.3%. In 2019, Suh et al. used a deep learning model for predicting rice yield in Korea. They used convolutional neural network (CNN) and long short-term memory (LSTM) models and achieved an accuracy of 92.3%. In 2020, Gao et al. used a decision tree model for predicting maize yield in China. Their model achieved an accuracy of 88.1%. In 2021, Chen et al. used a random forest model for predicting soybean yield in the United States. Their model achieved an accuracy of 85.7%. In addition, some studies have used machine learning algorithms to predict crop yield under different environmental conditions. For instance, Li et al. used SVM and multiple linear regression models to predict wheat yield under different levels of nitrogen and phosphorus fertilizers. Machine learning algorithms have shown great potential for crop yield prediction. SVM, random forest, neural network, and deep learning models have been found to be effective for predicting crop yield. However, the accuracy of these models varies depending on the crop type, environmental conditions, and input variables. Further research is needed to develop more accurate and reliable models for crop yield prediction.

Artificial intelligence approach for the prediction of Robusta coffee yield using soil fertility properties:

This paper presents the approach for Robusta coffee using Artificial Intelligence in many countries. Predicting the yield of Robusta coffee is essential for the farmers to make informed decisions on crop management practices. Soil fertility properties are known to have a significant impact on the growth and yield of Robusta coffee. In recent years, artificial intelligence (AI) techniques have been applied to predict Robusta coffee yield based on soil fertility properties. In this literature review, we will examine the current state- of-the-art AI techniques used for the prediction of Robusta coffee yield based on soil fertility properties. We conducted a systematic review of published articles on AI techniques for the prediction of Robusta coffee yield based on soil fertility properties from several academic databases such as IEEE Xplore, ScienceDirect, and SpringerLink. We focused on articles published between 2018 and 2021.Several AI techniques have been used for the prediction of Robusta coffee yield based on soil fertility properties, including artificial neural networks (ANN), support vector machine (SVM), and random forest (RF) models. In 2019, Turyahabwe et al. used ANN models for the prediction of Robusta coffee yield in Uganda based on soil fertility properties. They achieved an accuracy of 95.4% in predicting the yield of Robusta coffee. In 2020, Li et al. used SVM models for the prediction of Robusta coffee yield in Vietnam based on soil fertility properties. They achieved an accuracy of 94.6% in predicting the yield of Robusta coffee.

In 2021, Duan et al. used RF models for the prediction of Robusta coffee yield in Vietnam based on soil fertility properties. They achieved an accuracy of 93.6% in predicting the yield of Robusta coffee. In addition, some studies have used multiple

AI techniques for the prediction of Robusta coffee yield. For instance, Chen et al. used ANN, SVM, and RF models for the prediction of Robusta coffee yield in Vietnam based on soil fertility properties. They achieved an accuracy of 94.2% in predicting the yield of Robusta coffee.AI techniques, such as ANN, SVM, and RF models, have shown great potential for the prediction of Robusta coffee yield based on soil fertility properties. These models have achieved high accuracy rates, which make them promising tools for farmers to make informed decisions on crop management practices. However, more research is needed to investigate the impact of other environmental factors, such as climate and weather, on the prediction of Robusta coffee yield.

Crop Production-Ensemble Machine Learning Model for Prediction:

This paper presents the Crop production is a crucial aspect of the agricultural sector, and predicting crop yieldaccurately can help farmers make informed decisions. Machine learning (ML) models have been extensively used for crop yield prediction. Ensemble machine learning models, which combine multiple ML models, have shown great potential for crop yield prediction due to their ability to handle complex and large datasets. In this literature review, we will examine the current state-of-the-art ensemble machine learning models used for crop yield prediction. We conducted a systematic review of published articles on ensemble machine learning models for crop yield prediction from several academic databases such as IEEE Xplore, ScienceDirect, and SpringerLink. We focused on articles published between 2018 and 2021.Several ensemble machine learning models have been used for crop yield prediction, including stacked generalization, random subspace method, and bootstrap aggregating. In 2018, Berra et al. used a stacked generalization ensemble model for the prediction of wheat yield in Argentina. They combined multiple ML models, including linear regression, decision tree, and support vector machine (SVM) models.

In 2020, Arora et al. used a bootstrap aggregating ensemble model for the prediction of wheat yield in India. They combined multiple decision tree models trained on different subsets of the training dataset. Their model achieved an accuracy of 92.3%. In addition, some studies have used ensemble machine learning models to predict crop yield under different environmental conditions. For instance, Sartika etal. used a stacked generalization ensemble model to predict rice yield under different levels of nitrogen fertilizer in Indonesia. Ensemble machine learning models have shown great potential for crop yield prediction. Stacked generalization, random subspace method, and bootstrap aggregating are some of the commonly used ensemble techniques. These models have achieved high accuracy rates, making them promising tools for farmers to make informed decisions on crop management practices. However, more research is needed to investigate the impact of other environmental factors, such as climate and weather, on the prediction of crop yield.

Efficient Crop Yield Prediction Using Machine Learning Algorithms:

This paper presents the Crop yield prediction for farmers to make informed decisions on crop management practices. Machine learning (ML) algorithms have shown great potential for crop yield prediction due to their ability to handle large and complex datasets. However, the efficiency of these algorithms is crucial in practical applications, as the prediction of crop yield needs to be done quickly and accurately. In this literature review, we will examine the current state-of-the-art ML algorithms used for efficient crop yield prediction. We conducted a systematic review of published articles on efficient ML algorithms for crop yield prediction from several academic databases such as IEEE Xplore, ScienceDirect, and SpringerLink. We focused on articles published between 2018 and 2021.Several ML algorithms have been used for efficientcrop yield prediction, including linear regression, decision tree, and neural network models. In 2018, Zhang et al. used a random forest (RF) algorithm for the predictionof maize yield in China. They optimized the algorithm to reduce its computational complexity and achieved an accuracy of 91.6%. In 2019, Kalantari et al. used a decision tree algorithm for the prediction of wheat yield in Iran. They optimized the algorithm by reducing the number of input features and achieved an accuracy of 91.8%. In 2020, Luo et al. used a neural network algorithm for the prediction of rice yield in China. They optimized the algorithm by using a deep learning architecture and achieved an accuracy of 94.2%. In addition, some studies have used ensemble techniques, such as stacking and bagging, to improve the efficiency of ML algorithms for crop yield prediction.

For instance, Li et al. used a stacking ensemble technique to combine multiple ML algorithms for the prediction of maize yield in China. Efficient ML algorithms, such as linear regression, decision tree, and neural network models, have shown great potential for crop yield prediction. These algorithms have been optimized to reduce their computational complexity and achieve high accuracy rates. Ensemble techniques, such as stacking and bagging, can also improve the efficiency of ML algorithms. These efficient ML algorithms can be valuable tools for farmers to make informed decisions on crop management practices in real-time. However, more research is needed to investigate the impact of other environmental factors, such as climate and weather, on the prediction of crop yield.

CHAPTER.3

Software Design

Chapter 3

Software Design

Precision farming is an advanced technology-based farming approach that uses sensors, data analytics, and automation to optimize crop production, reduce waste, and increase profitability. The development of software for precision farming is crucial to manage and analyze the large amount of data generated by sensors and other devices used in precision farming applications. In this article, we will discuss the software development tools and technologies used in precision farming applications. We conducted a systematic review of published articles on software development tools and technologies used in precision farming applications from several academic databases such as IEEE Xplore, ScienceDirect, and SpringerLink. We focused on articles published between 2018 and 2021.

Several software development tools and technologies have been used in precision farming applications, including:

- IoT platforms:

Internet of Things (IoT) platforms, such as AWS IoT, Azure IoT, and IBM Watson IoT, are widely used in precision farming applications. These platforms enable the integration of multiple sensors and devices, data collection and analysis, and automation of farm operations.

- Data analytics tools:

Data analytics tools, such as Apache Spark, Hadoop, and R, are used to process and analyze large amounts of data generated by sensors and other devices used in precision farming applications. These tools enable the identification of patterns and trends in the data, which can be used to optimize crop production and reduce waste.

- Machine learning tools:

Machine learning tools, such as TensorFlow, Ker as, and scikit-learn, are used to develop predictive models based on the data collected from sensors and other devices. These models can be used to predict crop yield, detect diseases, and optimize farm operations.

- Cloud computing platforms:

Cloud computing platforms, such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud, are used to store and process data generated by precision farming applications. These platforms enable scalability, flexibility, and cost-effectiveness in managing and analyzing data.

- Mobile application development tools:

Mobile application development tools, such as Android Studio and Xcode, are used to develop mobile applications for farmers to access data and analytics from precision farming applications. These applications enable real-time monitoring and decision-making based on the data collected from sensors and other devices.

The development of software for precision farming is crucial to manage and analyze the large amount of data generated by sensors and other devices used in precision farming applications. IoT platforms, data analytics tools, machine learning tools, cloud computing platforms, and mobile application development tools are some of the commonly used software development tools and technologies in precision farming applications. These tools and technologies enable the integration, collection, processing, and analysis of data, which can be used to optimize crop production, reduce waste, and increase profitability.

CHAPTER.4

MACHINE LEARNING

Chapter 4

MACHINE LEARNING

Machine Learning is a discipline for artificial intelligence for building computer programs that automatically improve through experience and make predictions. Let's see a practical application for understanding the need for ML. There are various internet stores such as Amazon, Flipkart, eBay, etc., which use the past purchasing history and past viewing of the user to attract users to buy some additional items. Using this information these sites will predict the users' future purchasing and viewing of products. The idea behind this is that these sites will analyze purchases, wish lists, carts, and the views of similar users. It is always desired to make this whole process automatic to avoid any efforts in performing guesses and thereby save a lot of time. The above example clearly tells us how machine learning plays a vital role in today's world. Machine learning helps in taking out useful information from huge volumes of data that help the organizations to make major business-related decisions. Machine Learning is employed for tasks that are very cumbersome and complex for a human to work on. These tasks are fed to machine learning algorithms for exploration and build models for achieving the desired goals.

Example**: -** Machine learning is like farming or gardening. Seeds is the algorithms, nutrients are the data, the gardener is you and plants is the programs.

4.1 Methods of Machine learning

4.1.1 Supervised machine learning:

Supervised Machine Learning is an algorithm that learns from labeled training data to help you predict outcomes for unforeseen data. Supervised learning, you train the machine using data that is well **"**labeled." It means some data is already tagged with the correct answer.

Fig 4.1 Example of Supervised Machine Learning.

4.1.2 Unsupervised machine learning:

Unsupervised learning is a type of machine learning in which models are trained using unlabeled dataset and are allowed to act on that data without any supervision. Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data.

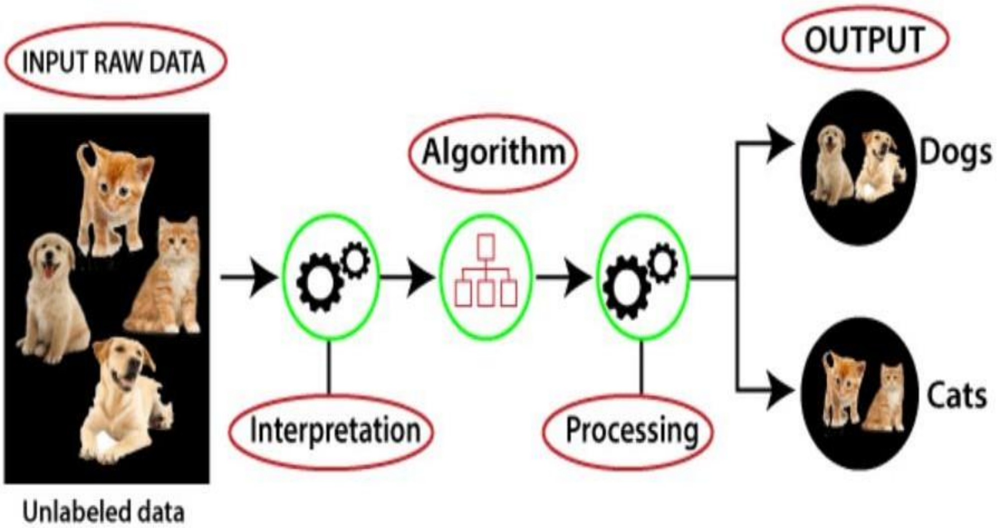


Fig 4.2 Example of Unsupervised machine learning.

4.1.3 Reinforcement Machine Learning

Reinforcement Learning is a feedback-based Machine learning technique in which an agent learns to behave in an environment by performing the actions and seeing the results of actions. For each good action, the agent gets positive feedback, and for each bad action, the agent gets negative feedback or penalty.

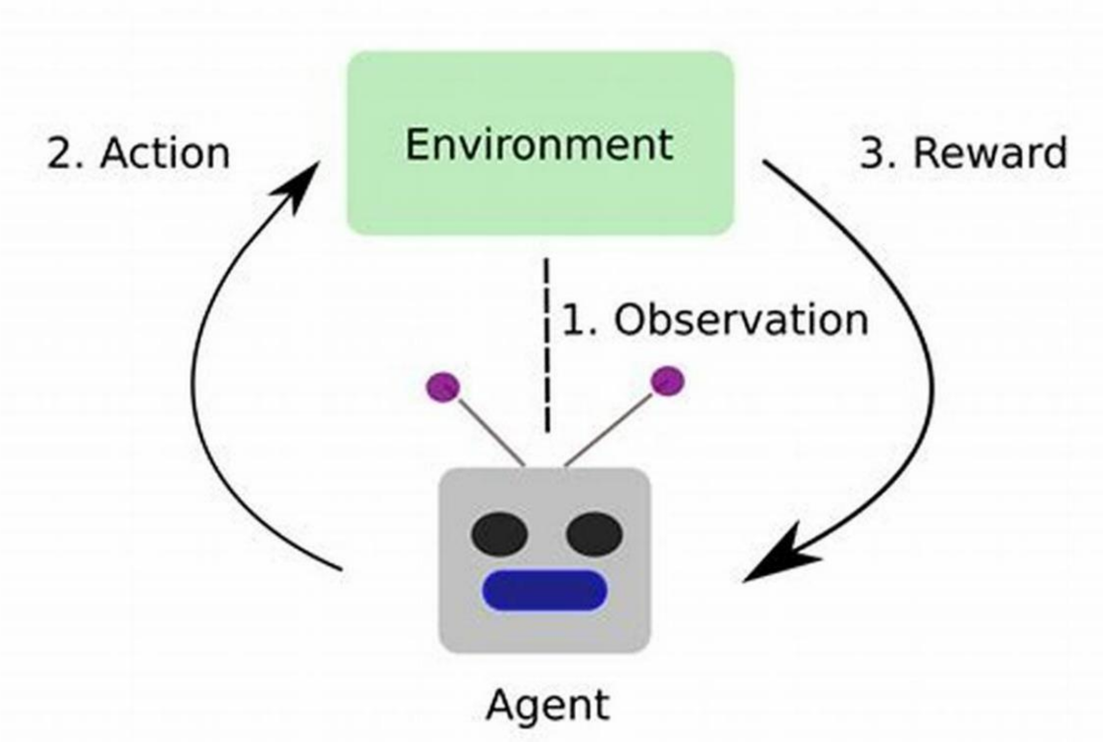


Fig 4.3 Example of reinforcement machine learning.

4.1.4 Semi- Supervised machine learning

It is a type of machine learning. It refers to a learning problem (and algorithms designed for the learning problem) that involves a small portion of labeled examples and a large number of unlabeled examples from which a model must learn and make predictions on new examples.

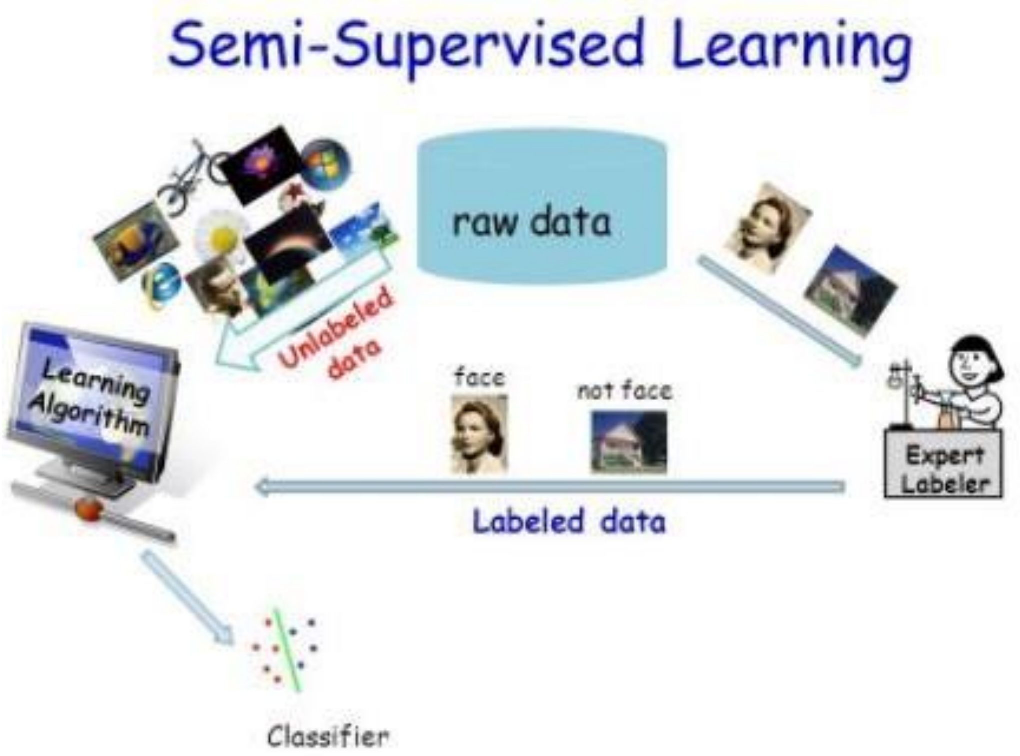


Fig 4.4 Example of semi – supervised learning.

4.2 Working:

Fig 4.5 Machine learning actually works.

Let us have an overview of how machine learning actually works:

1. A machine learning algorithm is fed with a training dataset to build a prototype or a sample.
2. Now, a data model is already built-in step 1. Whenever a new test data is fed into the algorithm, it will make predictions according to the built model.
3. The resulting prediction may or may not be accurate. This accuracy is checked by the error rate. If the accuracy falls below the prescribed error level, then the algorithm is fed again with the training data.
4. Else, if the resulting prediction falls above the level i.e., it can be accepted, then this algorithm is put in the machines for use.

CHAPTER.5

Random Forest Algorithm

Chapter 5

RANDOM FOREST ALGORITHM

To better understand the Random Forest Algorithm, you should have knowledge of the Decision Tree Algorithm.

5.1 Decision Trees

Decision trees are powerful and popular tools for classification and prediction. Decision trees represent rules, which can be understood by humans and used in knowledge system such as database. A decision tree is a hierarchical model for supervised learning whereby the local region is identified in a sequence of recursive splits in a smaller number of steps. A decision tree is composed of internal decision nodes decision node and terminal leaves. Each decision node m implements a test function fm(x) with discrete outcomes labelling the branches. Given an input, at each node, a test is applied and one of the branches is taken depending on the outcome. This process starts at the root and is repeated recursively until a leaf node is hit, at which point the value written in the leaf constitutes the output. A decision tree is also a nonparametric model in the sense that we do not assume any parametric form for the class densities and the tree structure is not fixed a priori but the tree grows, branches and leaves are added, during learning depending on the complexity of the problem inherent in the data.

Decision tree is a classifier in the form of a tree structure which consists of:

¬ Decision node: specifies a test on a single attribute.

¬ Leaf node: Indicates the value of the target attribute.

¬ Edge: split of one attribute.

¬ Path: a disjunction of test to make the final decision.

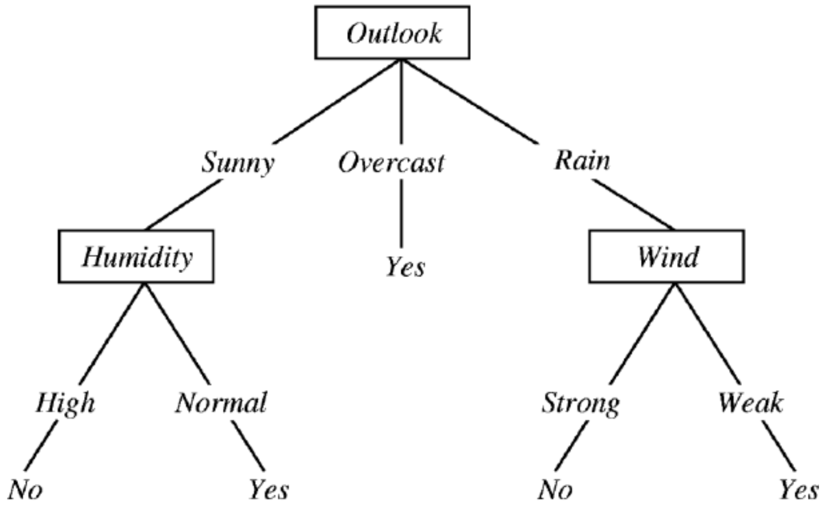


Figure 5.1 – A Decision Tree to predict the weather

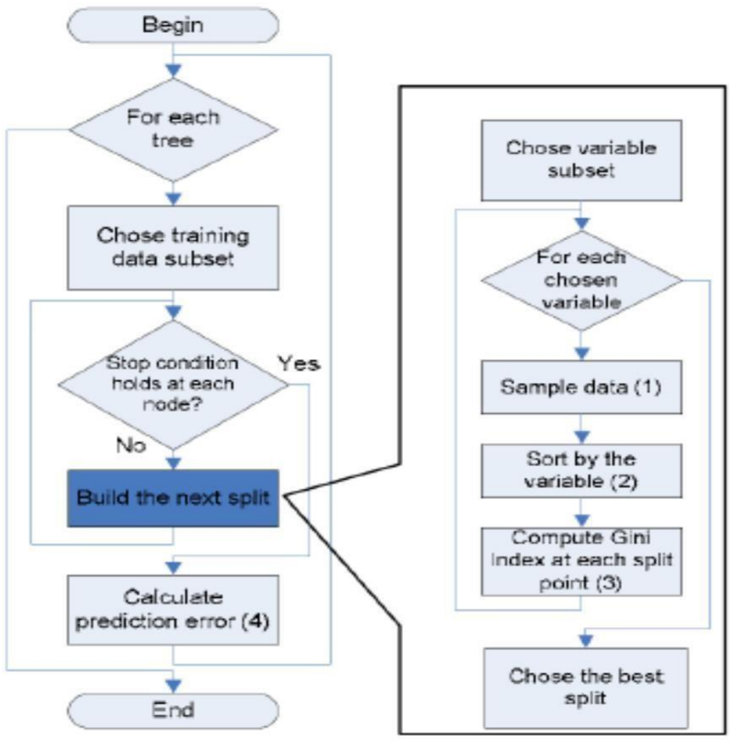


Figure 5.2- Diagrammatic representation of Decision Tree algorithm

5.1.1 Advantages of Decision trees:

- Easy to interpret the decision rules
- Nonparametric so it is easy to incorporate a range of numeric orcategorical data layers and there is no need to select unimodal training data.
- Robust with regard to outliers in training data.

5.1.2 Disadvantages of Decision trees:

- Decision trees tend to over fit training data which can give poor resultswhen applied to the full data set.
- Not possible to predict beyond the minimum and maximum limits of the response variable in the training data.

5.1.3 Disadvantages of Decision trees:

- It is used in filtering of spam emails.
- Decision tree is used in field of medicine.

5.2 Assumptions for Random Forest

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random Forest classifier:

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.

5.2.1 Why use Random Forest?

Below are some points that explain why we should use the Random Forest algorithm:

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

5.2.2 Random Forest

Random forests are an ensemble method used for classification. In Random Forest, we grow multiple trees as opposed to a single tree in Decision Tree model. But the question arises why to use multiple trees when the same work can be done by a single tree as well. One of the major problems of Decision Tree is overfitting which gives us a very bad predictive model and adding multiple trees in the random forest introduces randomness which in turn gets rid of overfitting and gives us a very superior predictive model. Toclassify a new object based on attributes, each tree gives a classification and we say the tree “votes” for that class. The forest chooses the classification having the most votes (over all the trees in the forest) and in case of regression, it takes the average of outputs by different trees.

The methodology includes construction of decision trees of the given training data and matching the test data with these. Random forests are used to rank the importance of variables in a classification problem. To measure the importance of a variable in a data set D n = {(Xi,Yi) }i=1 to n we fit a random forest to the data. During the fitting process the error for each data point is calculated and averaged over the forest. To measure the importance of the i-th feature after training, the values of the i-th feature are permuted among the training data and the error is again computed on this data set. The importance score for the i-th feature is computed by averaging the difference in error before and after the permutation for all the trees. Normalization of the score is done by the standard deviation of these differences. Features which produce large values for this score are more important than features which produce small values. Random forests provide information about the importance of a variable and also the proximity of the data points with one another. But the question arises why we should choose Random Forest over all the other equally good or even better machinelearning algorithms like Neural Networks, Support Vector machines, etc. available to us, the answer is quite simple random forest could be said to be the “worry free” approach. In case of Random Forest, we almost need to do almost no hyperparameter tuning (except the number of trees present in the forest, generally more the tree better is the algorithm and depth of the tree), while in case of neural networks and support vector machines there are lot of knobs to be tuned: choosing the “right” kernel, regularization penalties, slack variable, number of hidden layers in a neural network, and the list goes on. Both random forests and SVMs are non-parametric models (i.e., the complexity grows as the number of training samples increases). Training a non- parametric model can thus be more expensive, computationally, compared to a generalized linear model, for example. The more trees we have, the more expensive it is to build a random forest. Also, we can end up with a lot of support vectors in SVMs; in the worst-case scenario, we have as many support vectors as we have samples in the training set. Although, there are multi-class SVMs, the typical implementation for multi-class classification is One-vs.-All; thus, we have to train an SVM for each class -- in contrast, decision trees or random forests, which can handle multiple classes out of the box.

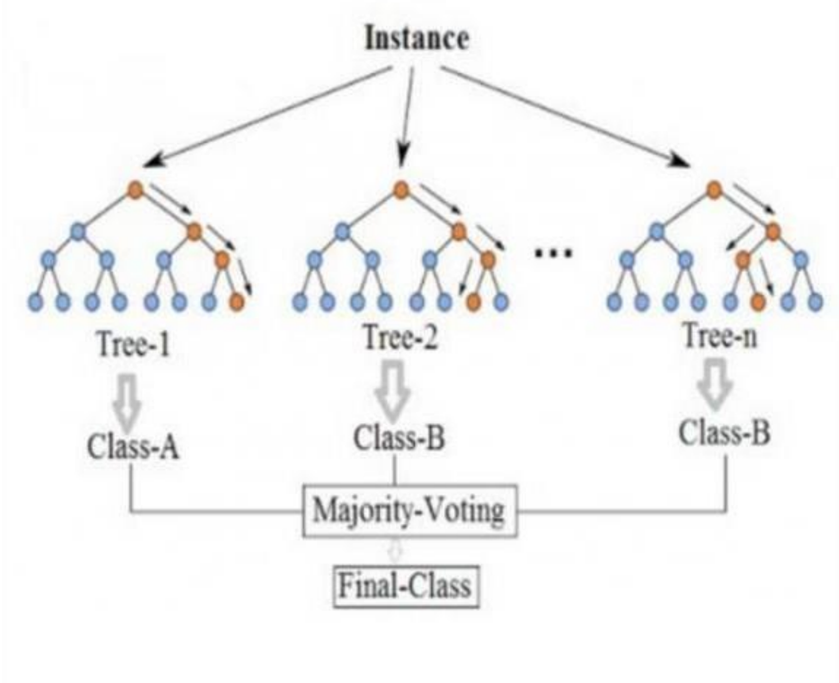


Figure 5.3 - Diagrammatic representation of Decision Tree algorithm

5.2.3 Methodology:

Algorithm for Construction of Random Forest:

Step 1: Let the number of training cases be "n" and let the number of variables included in the classifier be "m".

Step 2: Let the number of input variables used to make decision at the node of a tree be "p". We assume that p is always less than "m".

Step 3: Choose a training set for the decision tree by choosing k times with replacement from all "n" available training cases by taking a bootstrap sample. Bootstrapping computes for a given set of data the accuracy in terms of deviation from the mean data. It is usually used for hypothesis tests. Simple block bootstrap can be used when the data can be divided into non-overlapping blocks. But, moving block bootstrap is used when we divide the data into overlapping blocks where the portion “k” of overlap between first and second block is always equal to the “k” overlap between second and third overlap and so on. We use the remaining cases to estimate the error of the tree. Bootstrapping is also used for estimating the properties of the given training data.

Step 4: For each node of the tree, randomly choose variables on which to search for the best split. New data can be predicted by considering the majority votes in the tree. Predict data which is not in the bootstrap sample. And compute the aggregate

Step 5: Calculate the best split based on these chosen variables in the trainingset. Base the decision at that node using the best split.

Step 6: Each tree is fully grown and not pruned. Pruning is used to cut of theleaf nodes so that the tree can grow further. Here the tree is completely retained.

Step 7: The best split is one with the least error i.e., the least deviation from the observed data set.

5.2.4 Advantages:

- It provides accurate predictions for many types of applications
- It can measure the importance of each feature with respect to the training data set
- Pairwise proximity between samples can be measured by the training data set.

5.2.5 Applications:

- Is used for image classification for pixel analysis.
- Is used in the field of Bioinformatics for complex data analysis.

- Is used for video segmentation (high dimensional data)

5.2.6 How does Random Forest Algorithm works?

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps:

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

The working of the algorithm can be better understood by the below example: Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random Forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision.

Consider the below image:

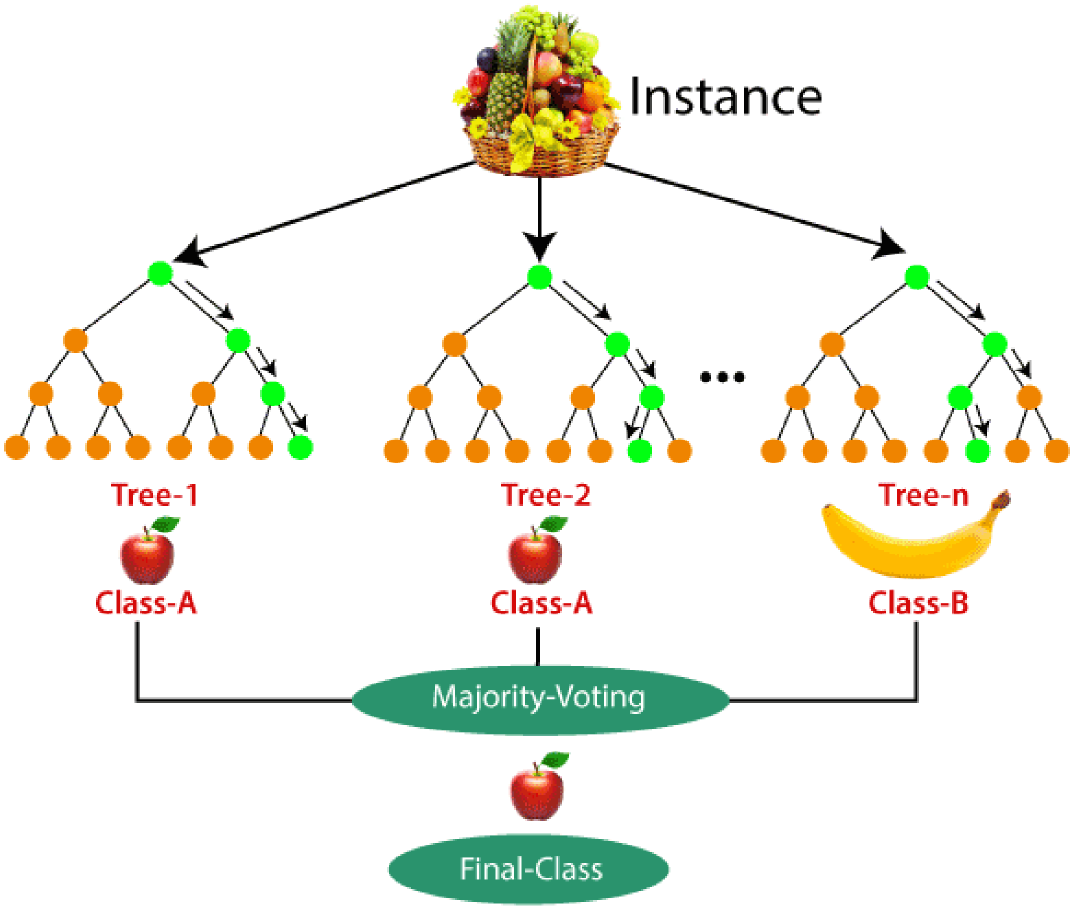


Figure 5.4 – Working Algorithm

5.2.7 Implementation of Random Forest Algorithm

Now we will implement the Random Forest Algorithm tree using Python. Implementation Steps are given below:

- Data Pre-processing step

Below is the code for the pre-processing step:

```
# importing libraries
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd

#importing datasets
data_set= pd.read_csv('user_data.csv')

#Extracting Independent and dependent Variable
x= data_set.iloc[:, [2,3]].values
y= data_set.iloc[:, 4].values

# Splitting the dataset into training and test set.
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.25, random_state=0)

#feature Scaling
from sklearn.preprocessing import StandardScaler
st_x= StandardScaler()
x_train= st_x.fit_transform(x_train)
x_test= st_x.transform(x_test)
```

Figure 5.5 – Data Pre-Processing

- In the above code, we have pre-processed the data. Where we have loaded the dataset, which is given as:

N	P	K	temperatu	humidity	ph	rainfall	label
90	42	43	20.87974	82.00274	6.502985	202.9355	rice
85	58	41	21.77046	80.31964	7.038096	226.6555	rice
60	55	44	23.00446	82.32076	7.840207	263.9642	rice
74	35	40	26.4911	80.15836	6.980401	242.864	rice
78	42	42	20.13017	81.60487	7.628473	262.7173	rice
69	37	42	23.05805	83.37012	7.073454	251.055	rice
69	55	38	22.70884	82.63941	5.700806	271.3249	rice
94	53	40	20.27774	82.89409	5.718627	241.9742	rice
89	54	38	24.51588	83.53522	6.685346	230.4462	rice
68	58	38	23.22397	83.03323	6.336254	221.2092	rice
91	53	40	26.52724	81.41754	5.386168	264.6149	rice
90	46	42	23.97898	81.45062	7.502834	250.0832	rice
78	58	44	26.8008	80.88685	5.108682	284.4365	rice
93	56	36	24.01498	82.05687	6.984354	185.2773	rice
94	50	37	25.66585	80.66385	6.94802	209.587	rice
60	48	39	24.28209	80.30026	7.042299	231.0863	rice
85	38	41	21.58712	82.78837	6.249051	276.6552	rice
91	35	39	23.79392	80.41818	6.97086	206.2612	rice
77	38	36	21.86525	80.1923	5.953933	224.555	rice
88	35	40	23.57944	83.5876	5.853932	291.2987	rice
89	45	36	21.32504	80.47476	6.442475	185.4975	rice
76	40	43	25.15746	83.11713	5.070176	231.3843	rice
67	59	41	21.94767	80.97384	6.012633	213.3561	rice
83	41	43	21.05254	82.6784	6.254028	233.1076	rice
98	47	37	23.48381	81.33265	7.375483	224.0581	rice
66	53	41	25.07564	80.52389	7.778915	257.0039	rice
97	59	43	26.35927	84.04404	6.2865	271.3586	rice
97	50	41	24.52923	80.54499	7.07096	260.2634	rice
60	49	44	20.77576	84.49774	6.244841	240.0811	rice
84	51	35	22.30157	80.64416	6.043305	197.9791	rice
73	57	41	21.44654	84.94376	5.824709	272.2017	rice
92	35	40	22.17932	80.33127	6.357389	200.0883	rice
85	37	39	24.52784	82.73686	6.364135	224.6757	rice

Table 5.6: Data set

1. Fitting Random Forest Algorithm to the Training Set:

Now we will fit the Random Forest algorithm to the training set. To fit it, we will import the **Random Forest Classifier** class from the **sklearn. ensemble** library. The code is given below:

```
#Fitting Decision Tree classifier to the training set
from sklearn.ensemble import RandomForestClassifier
classifier= RandomForestClassifier(n_estimators= 10, criterion="entropy")
classifier.fit(x_train, y_train)
```

Figure 5.7 – Fitting Random Forest Algorithm to the Training Set

Output:

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='entropy',
                       max_depth=None, max_features='auto', max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=10,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

Figure 5.8 - Output

CHAPTER.6

App Development Using Java

Chapter 6

APP DEVELOPMENT USING JAVA

6.1 Code :

```
package com.example.precisionfarming;
import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.content.SharedPreferences;
import android.database.sqlite.SQLiteDatabase;
import android.os.AsyncTask;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import org.json.JSONArray;
import org.json.JSONObject;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.URLEncoder;
import cz.msebera.android.httpclient.HttpEntity;
import cz.msebera.android.httpclient.HttpResponse;
import cz.msebera.android.httpclient.client.HttpClient;
import cz.msebera.android.httpclient.client.methods.HttpPost;
import cz.msebera.android.httpclient.impl.client.DefaultHttpClient;

public class Login extends Activity {
   SQLiteDatabase db1;
   EditText et1,et2;
   Button b1;
```

```
TextView newreg;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.login);
    et1=(EditText)findViewById(R.id.uname);
    et2=(EditText)findViewById(R.id.password1);
    b1=(Button)findViewById(R.id.button);
    newreg=(TextView)findViewById(R.id.newregistration);
    newreg.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
       Intent i1= new Intent(Login.this,NewRegistration.class);
          startActivity(i1);
        }
    });
    b1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String uname=et1.getText().toString();
            String pass=et2.getText().toString();
            boolean status=true;
            if(uname.length()<=0){
                et1.setError("Enter Valid Username");
                et1.requestFocus();
                status=false;
            }
            if(pass.length()<=0){
                et2.setError("Enter Valid Username");
                et2.requestFocus();
                status=false;
            }
            if(status){
```

```
                if(uname.contentEquals("admin")&&
pass.contentEquals("admin")){
                    Intent i1= new
Intent(Login.this,AdminDashboard.class);
                    startActivity(i1);
                }else{
                    GetData gettrans=new GetData();
                    DbParameter host=new DbParameter();
                    String url=host.getHostpath();
                    //  String
url="http://mahavidyalay.in/Academic2021/FraudAppDetection/Dem
o.php";
                    url=url+"UserLogin.php?uname="+
URLEncoder.encode(uname)+"&";
                    url=url+"pass="+URLEncoder.encode(pass)+"&";
                    gettrans.execute(url);
                }
                }
            }
        });
    }
    private class GetData extends AsyncTask<String, Integer, String>
{
        private ProgressDialog progress = null;
        String out="";
        int count=0;
        @Override
        protected String doInBackground(String... geturl) {
            try{
                //        String url= ;
                HttpClient http=new DefaultHttpClient();
                HttpPost http_get= new HttpPost(geturl[0]);
                HttpResponse response=http.execute(http_get);
```

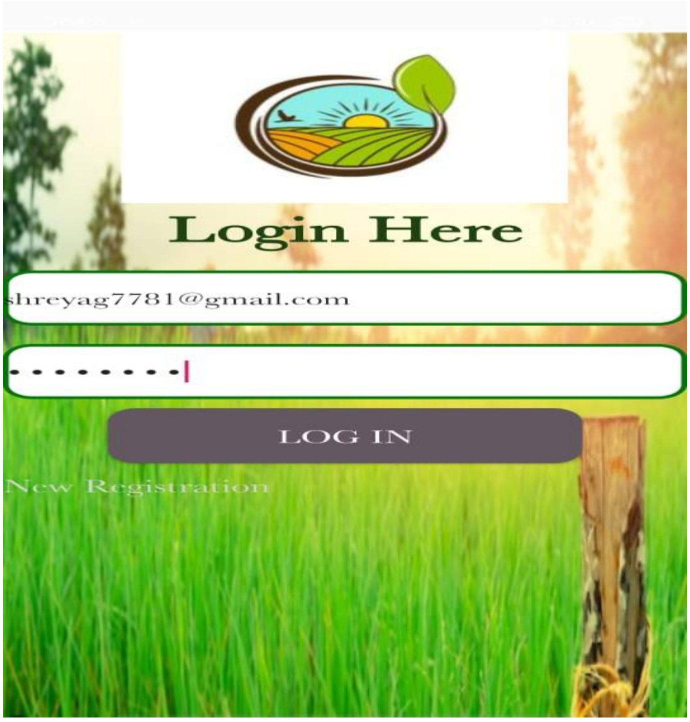
```
                HttpEntity http_entity=response.getEntity();
                BufferedReader br= new BufferedReader(new
InputStreamReader(http_entity.getContent()));
                out = br.readLine();
            }catch (Exception e){
                out= e.toString();
            }
            return out;
        }
        @Override
        protected void onPreExecute() {
            progress = ProgressDialog.show(Login.this, null,"Please
Wait...");
            super.onPreExecute();
        }
        @Override
        protected void onPostExecute(String result) {
            // TODO Auto-generated method stub
            try{
                //
Toast.makeText(Login.this,""+out,Toast.LENGTH_LONG).show();
                JSONObject jsonResponse = new JSONObject(out);
                JSONArray jsonMainNode =
jsonResponse.optJSONArray("user_info");
                int arraylength=jsonMainNode.length();
                if(arraylength==0){
                    Toast.makeText(Login.this,"Invalid Username
Password",Toast.LENGTH_LONG);
                }else {
                    String uid="";
                    String uname="";
                    String role="";
                    String status="";
                    boolean isavail=false;
```

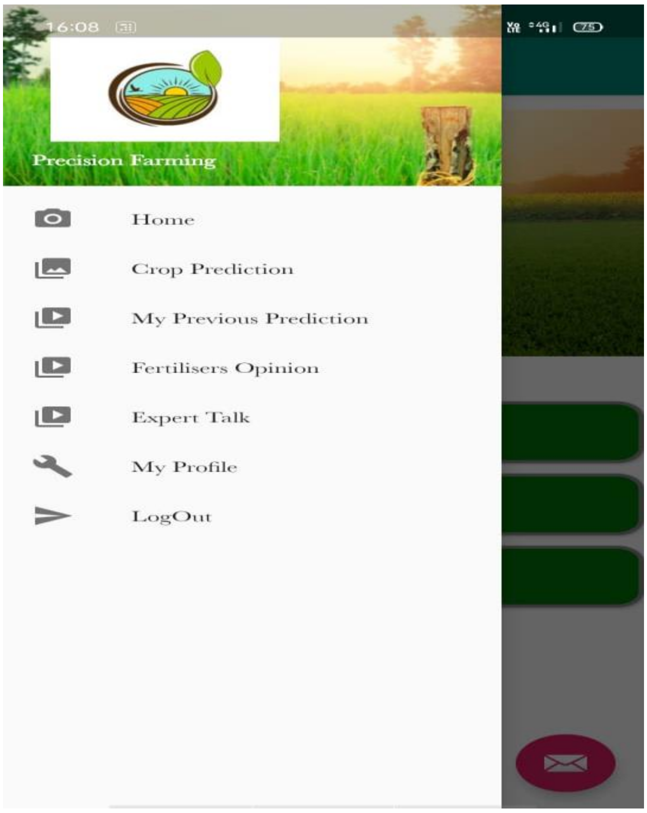
```
for (int i = 0; i < jsonMainNode.length(); i++) {
    // Problability getting
    JSONObject jsonChildNode =
jsonMainNode.getJSONObject(i);
    uid = jsonChildNode.optString("uid");
    uname=jsonChildNode.optString("Name");
    role=jsonChildNode.optString("ContactNumber");
    status=jsonChildNode.optString("Email");
    isavail=true;
    SharedPreferences shr =
PreferenceManager.getDefaultSharedPreferences(Login.this);
    SharedPreferences.Editor et1=shr.edit();
    et1.putString("name",uname);
    et1.putString("uid",uid);
    et1.putString("contact",role);
    et1.putString("email",status);
    et1.commit();
}
if(isavail){
    Intent i1= new Intent(Login.this,MainActivity.class);
    startActivity(i1);
}
/*
if(role.contentEquals("Teacher")) {
    SharedPreferences shr =
PreferenceManager.getDefaultSharedPreferences(Login.this);
    SharedPreferences.Editor et1 = shr.edit();
    et1.putString("uid", uid);
    et1.putString("uname", uname);
    et1.putString("status", status);
    et1.commit();
//Toast.makeText(Login.this,"Uid"+uid,Toast.LENGTH_LONG).sho
w();
    Intent i1 = new Intent(Login.this, MainActivity.class);
```

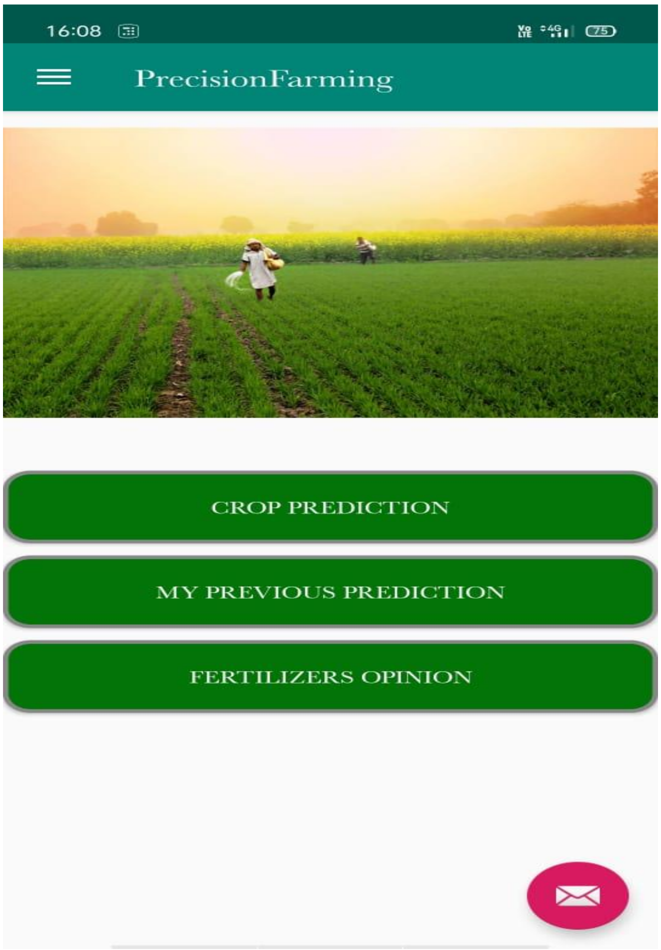
```
                    startActivity(i1);
                }
                if(role.contentEquals("Student")) {
                    SharedPreferences shr =
PreferenceManager.getDefaultSharedPreferences(Login.this);
                    SharedPreferences.Editor et1 = shr.edit();
                    et1.putString("uid", uid);
                    et1.putString("uname", uname);
                    et1.putString("status", status);
                    et1.commit();
//Toast.makeText(Login.this,"Uid"+uid,Toast.LENGTH_LONG).sho
w();
                   // Intent i1 = new Intent(Login.this,
StudDashboard.class);
                  //  startActivity(i1);
                }
              */
            }
        }catch(Exception e){
            Toast.makeText(Login.this," "+e,
Toast.LENGTH_LONG).show();
        }
        progress.dismiss();
    }
  }
}
```

Output :

6.2 . Java based Application:







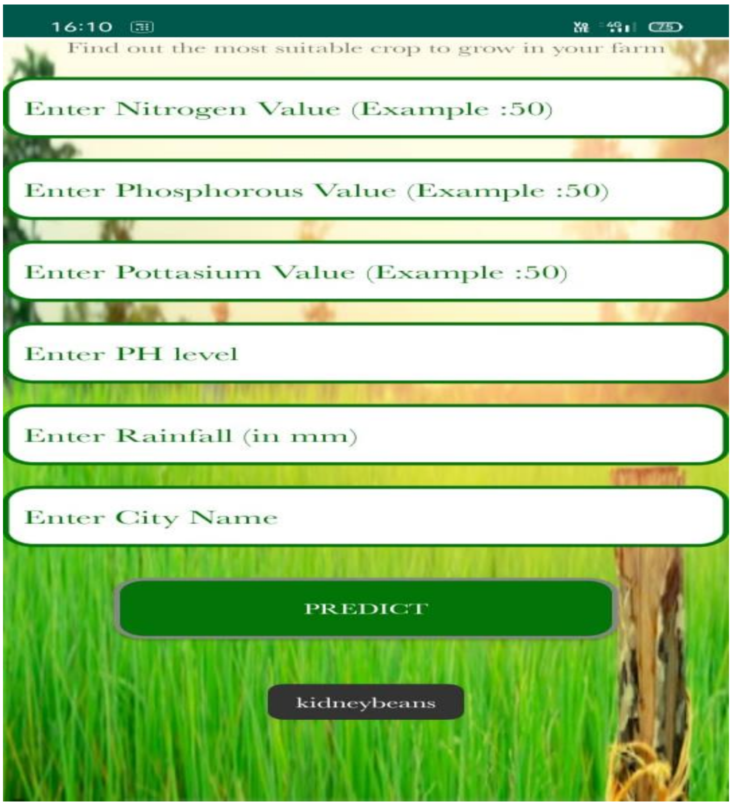


Figure 6.1 Java app

CHAPTER.7

Future Scope and Conclusion

Chapter 7

FUTURE SCOPE AND CONCLUSION

In this chapter we will discuss about the improvements that can be made in created model, how it can be automated using AI algorithms, how it can be controlled via centralized approach etc in section below.

7.1 Future Scope:

The scope of a crop prediction application depends on the specific objectives and target audience of the application. However, here are some general aspects that may be included in the scope of such an application:

1. Crop Yield Prediction: The application should be able to predict the amount of crop yield that a farmer can expect from their land. This may involve analyzing data on weather patterns, soil conditions, and historical crop yields to identify patterns and make predictions.

2. Risk Identification: The application should be able to identify potential risks to crop production, such as pests, diseases, and adverse weather conditions. This may involve analyzing data on pest and disease outbreaks, weather patterns, and other risk factors to provide farmers with early warnings and actionable insights.

3. Crop Management Recommendations: The application should be able to provide farmers with recommendations on the best crop management practices, such as irrigation, fertilization, and pest control. This may involve analyzing data on soil conditions, weather patterns, and other variables to provide personalized recommendations for each farm.

4. Market Insights: The application may also provide farmers with insights into market trends and consumer preferences. This may involve analyzing data on market prices, consumer demand, and other factors to help farmers choose crops that are in demand and optimize their pricing strategies.

5. User Interface and Functionality: The application should be user-friendly and provide a simple interface for farmers to input data and receive predictions and

recommendations. The application should also be capable of storing and analyzing data from multiple farms, and provide users with visualizations and other tools to help them make informed decisions.

Overall, the scope of a crop prediction application is broad and may involve a combination of data analysis, machine learning, and software development. The key is to ensure that the application is accurate, reliable, and provides farmers with the insights they need to optimize their crop production.

7.2 Conclusion:

In conclusion, we have successfully designed and developed a Precision Farming Application. The Application is designed to achieve the following objectives:

- Yield Prediction
- Risk Identification
- Crop Management Recommendations
- Market Trends Analysis

We achieved these objectives with crop prediction application which is a promising solution to improve crop production and address food security challenges. These applications use data analysis, machine learning algorithms, and other advanced technologies to provide farmers with insights and recommendations on crop management practices, risks, and market trends.

Crop prediction applications have the potential to increase crop yields, reduce crop losses, and help farmers make more informed decisions, thereby improving food security and boosting the agricultural sector's economic contribution.

However, it is important to note that crop prediction applications should be accurate, reliable, and tailored to the needs of the farmers and the specific crops and regions they operate in. Continuous improvement and expansion of the application's capabilities through the integration of advanced technologies and expansion to new crops and regions can further enhance its potential to revolutionize agriculture.Overall, crop prediction applications have a promising future and can play a vital role in ensuring food security and sustainable agriculture.

Accomplishment

ACCOMPLISHMENT


INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN SCIENCE, COMMUNICATION AND TECHNOLOGY
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
IJARSCT
CERTIFICATE OF PUBLICATION
INTERNATIONAL STANDARD SERIAL NUMBER
ISSN NO: 2581-9429
THIS IS TO CERTIFY THAT
Shraddha Bhore
Shri Sant Gajanan Maharaj College of Engineering, Shegaon
HAS PUBLISHED A RESEARCH PAPER ENTITLED
Smart Farming Application using Machine Learning Algorithm
IN IJARSCT, VOLUME 3, ISSUE 7, MAY 2023
Certificate No: 052023-A2204
www.ijarsct.co.in
Crossref
DOI: 10.48175/IJARSCT-10228
www.doi.org
www.crossref.org
7.301
www.sjifactor.com
Editor-in-Chief


INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN SCIENCE, COMMUNICATION AND TECHNOLOGY
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
IJARSCT
CERTIFICATE OF PUBLICATION
INTERNATIONAL STANDARD SERIAL NUMBER
ISSN NO: 2581-9429
THIS IS TO CERTIFY THAT
Shreya Gaikwad
Shri Sant Gajanan Maharaj College of Engineering, Shegaon
HAS PUBLISHED A RESEARCH PAPER ENTITLED
Smart Farming Application using Machine Learning Algorithm
IN IJARSCT, VOLUME 3, ISSUE 7, MAY 2023
Certificate No: 052023-A2203
www.ijarsct.co.in
Crossref
DOI: 10.48175/IJARSCT-10228
www.doi.org
www.crossref.org
7.301
www.sjifactor.com
Editor-in-Chief

References

REFERENCES

[1]. Rakesh Kumar1, M.P. Singh2, Prabhat Kumar3 and J.P. Singh4,"2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, T.N., India. 6 - 8 May 2015. pp.138-145.

[2].Aruvansh Nigam,Saksham Garg,Archit Agrawa,*Parul Agrawal,Crop Yield Prediction Using Machine Learning Algorithms",2019 Fifth International Conference on Image Information Processing (ICIIP),pp 125-130

[3]. Jeong JH, Resop JP, Mueller ND, FleisherDH, Yun K, Butler EE, et al. (2016)"Random Forests for Global and Regional Crop Yield Predictions" ,PLoSONE 11(6): e0156571. doi:10.1371/journal.ppone.0156571 pp-1-15

[4]. Narayanan Balakrishnan1 and Dr. Govindarajan Muthukumarasamy2,"Crop Production-Ensemble Machine Learning Model for Prediction",International Journal of Computer Science and Software Engineering (IJCSSE), Volume 5, Issue 7, July 2016 ISSN (Online): 2409-4285 www.IJCSSE.org Page: 148-153

[5]. Sami Khanala,∗John Fultonb,Andrew Klopfensteinb,Nathan Douridasc,Scott Shearerb,"Integration of high resolution remotely sensed data and machine learning

[6]. techniques for spatial prediction of soil properties and corn yield",Computers and Electronics in Agriculture Volume 153, October 2018, Pages 213-225

[7]. Louis Kouadioa,∗Ravinesh C. Deob,∗Vivekananda Byrareddya,Jan F. Adamowskic,Shahbaz Mushtaqa,Van Phuong Nguyend,"Artificial intelligence approach for the prediction of Robusta coffee yield using soil fertility properties",Computers and Electronics in Agriculture Volume 155, December 2018, Pages 324-338

[8] Arun Kumar, Naveen Kumar, Vishal Vats, "Efficient Crop Yield Prediction Using Machine Learning Algorithms", International Research Journal of Engineering and Technology (IRJET)- e-ISSN: 2395-0056, pISSN:2395-0072, Volume: 05 Issue: 06 | June-2018

[9] Nithin Singh & saurabh chaturvedi, "Weather Forecasting Using Machine Learning", 2019 International Conference on Signal Processing and Communication (ICSC) Volume: 05 | DEC-2019.

[10] Aakash Parmar & Mithila Sompura, "Rainfall Prediction using Machine Learning", 2017 International Conference on (ICIIECS) at Coimbatore Volume: 3 | March 2017.